Design and MIMO Control of A Hyper-Redundant Robotic Arm

Xingsheng Xu

Advisor: Raúl Ordóñez

Department of Electrical and Computer Engineering, University of Dayton

Key Words

- **Hyper-redundant robots (HRR)** are a kind of robots that have a very large degree of kinematic redundancy.
- **Forward kinematics** is an arithmetic operation, which determines the position and orientation of the end-effector given the values of each joint parameter of the robot using the kinematic equations.
- **Multi-input multi-output (MIMO) control approach** inputs torque of each joint to control joint dynamic variables such as position, orientation, velocity and acceleration in a hyper-redundant robotic system.

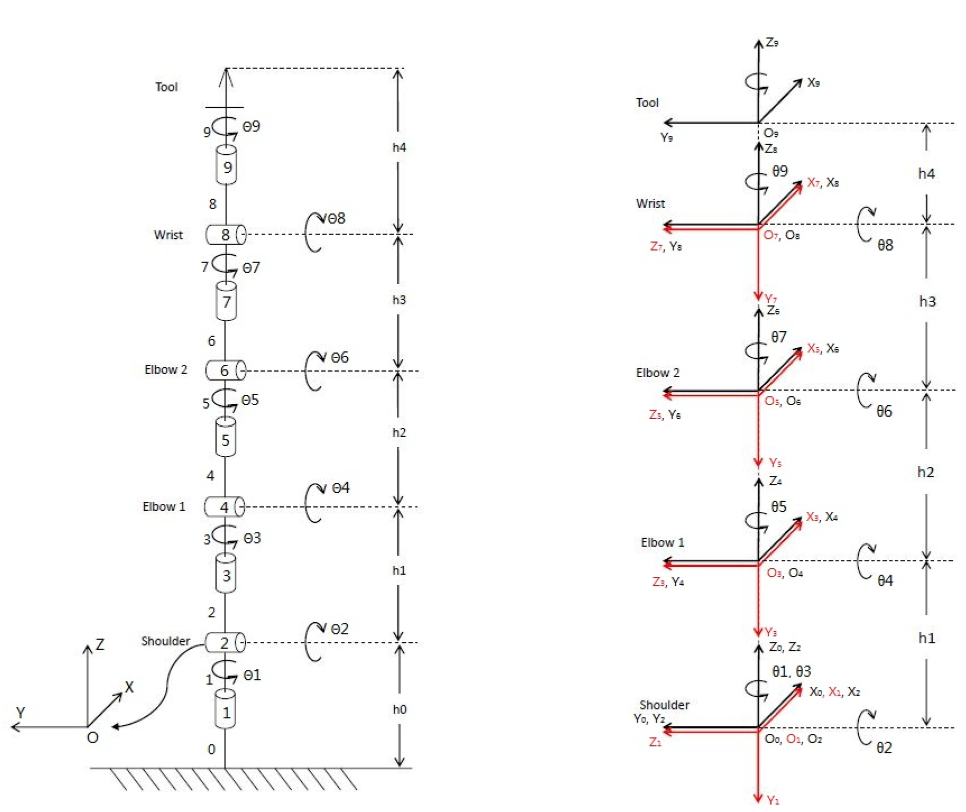
(a) Snake (b) Elephant Trunk (c) Tentacle

Figure 1: Hyper-redundant robots.

Motivation

- An application **robotic platform** has been constructed based on both the kinematic and dynamic model of a hyper-redundant manipulator.
- Also, we apply the **MIMO input-output feedback linearization control approach** to this hyper-redundant robotic system to control the end-effector in Cartesian space. This control approach can highly improve the robotic performance while executing motion control or tracking a path in a constrained and complex environment.

Kinematic Model



(a) Joint schematic (b) Frame assignment

This model is established at its home position. The arm is composed of a set of revolute joints which are arranged orthogonal to each other while making the shoulder and wrist joint spherical.

Performance without Dynamic Control

- **Tracking A Rectangle on X-Y Plane**:

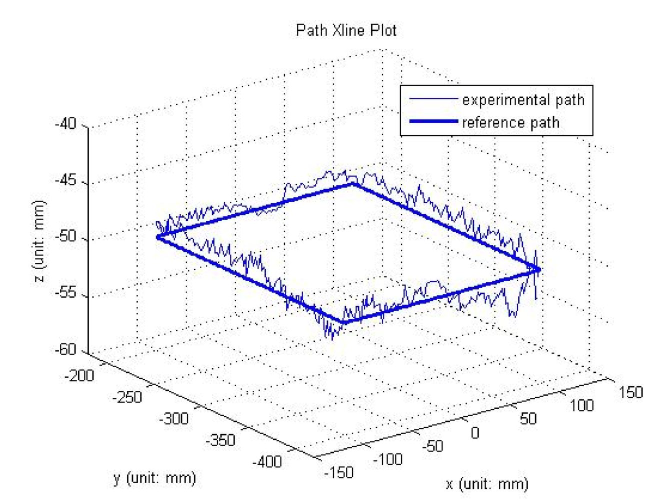


Figure 2: Tracking-trajectory of a rectangle on X-Y plane.

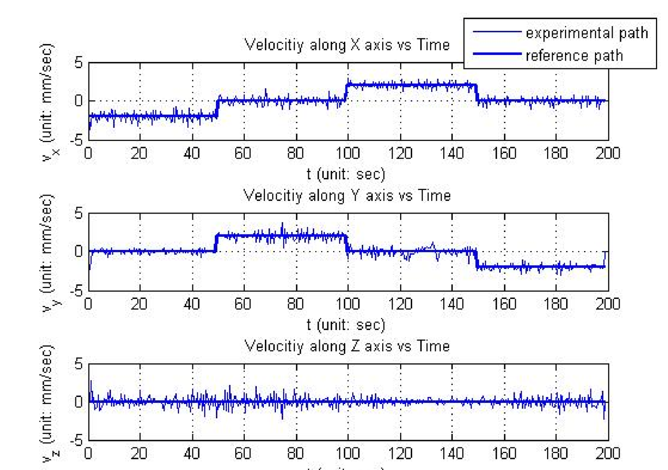


Figure 3: Velocity of tracking-trajectory of a rectangle on X-Y plane.

- **Tracking An Inclined Circle**:

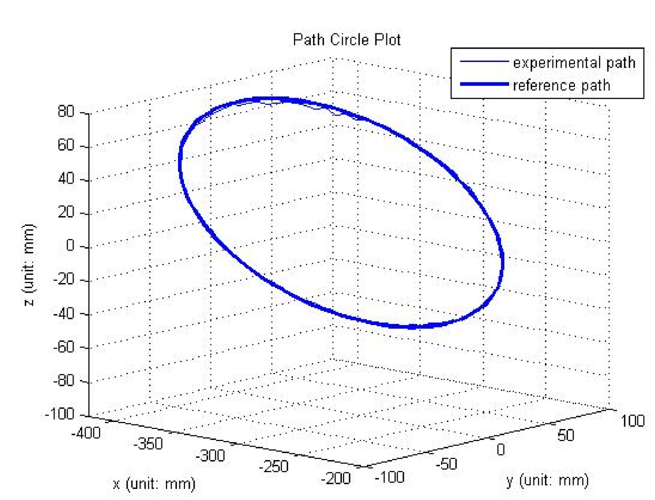


Figure 4: Tracking-trajectory of an inclined circle.

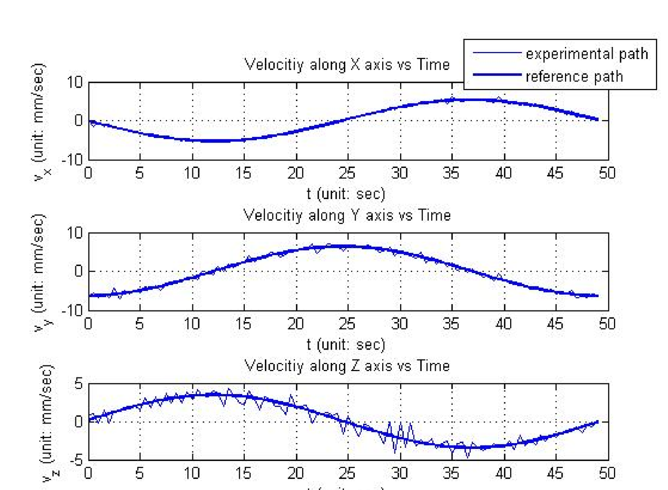


Figure 5: Velocity of tracking-trajectory of an inclined circle.

Dynamic Model

- **Manipulator Jacobian Matrices**: An expressions to connect the relationship between angular velocity ω_n^0, linear velocity v_n^0 of the end-effector and joint velocity $\dot{q}$ as

$$\omega_n^0 = J_\omega \dot{q},$$
$$v_n^0 = J_v \dot{q},$$

where J_ω and J_v are $3 \times n$ matrices.

- **Euler-Lagrange Equation**: An application leads to a robotic system of n coupled, second order nonlinear ordinary differential equations of the form

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \tau_i, i = 1, ..., n,$$

where τ_i is input torque of each motor and the the Lagrangian L is given by

$$L = K - P,$$

where K is the kinetic energy and P is the potential energy.

Algorithm of MIMO Control

- **Joint Space MIMO Control**: We can rewrite the system dynamics from the Euler-Lagrange equation into Input-Output form as

$$\ddot{q}_i = f(q_i, \dot{q}_i) + g(q_i, \dot{q}_i) \cdot \tau_i,$$

and outputs

$$y_i = q_i, i = 1, ..., n.$$

Then, we pick our MIMO controllers as

$$\tau_i = g(q_i, \dot{q}_i)^{-1} \cdot [-f(q_i, \dot{q}_i) + v_i], i = 1, ..., n.$$

The output tracking errors will be

$$e_i = q_i - r_i,$$

where r_i is joint variable reference. We set $v_i = -k_i e_i - k_{i+1}\dot{e}_i$ and get outputs $y_i = -k_i e_i - k_{i+1}\dot{e}_i$.

- **Cartesian Space MIMO Control**: The Jacobian matrix can be a smooth and invertible mapping between the joint variables $q \in Q$ and the workspace variables $\Gamma \in \mathbb{R}^n$ as

$$\dot{\Gamma} = J\dot{q}.$$

We take the derivative of both sides of the equation, then substitute $\ddot{q}$ created by the joint space to get

$$\ddot{\Gamma} = F(q_i, \dot{q}_i) + G(q_i, \dot{q}_i) \cdot \tau_i,$$

where $F(q_i, \dot{q}_i) = J \cdot f(q_i, \dot{q}_i) + \dot{J}\dot{q}$ and $G(q_i, \dot{q}_i) = J \cdot g(q_i, \dot{q}_i)$. After that, we can derive our MIMO workspace controllers as

$$\tau_i = G(q_i, \dot{q}_i)^{-1} \cdot [-F(q_i, \dot{q}_i) + v_i], i = 1, ..., n.$$

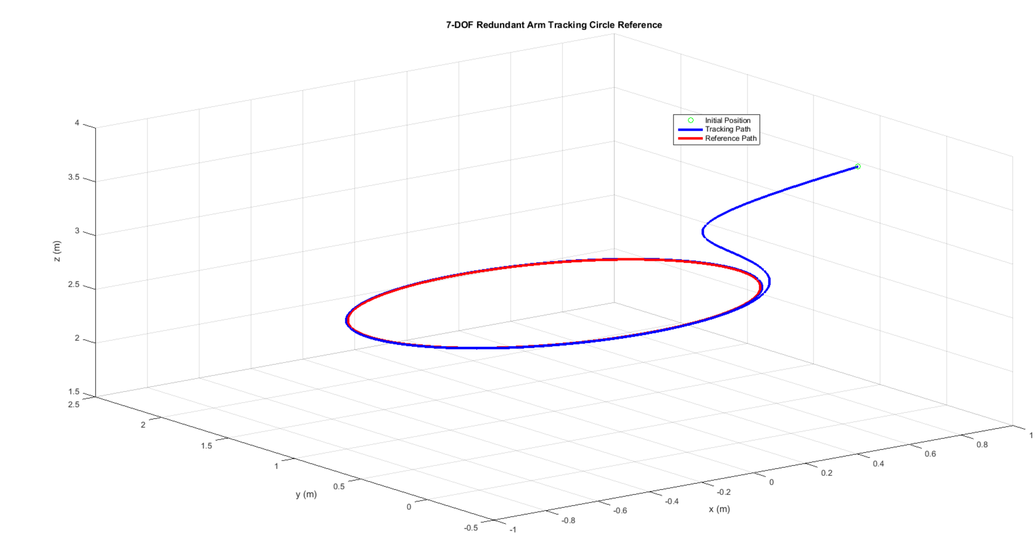
The output tracking errors will be

$$e_i = \Gamma_i - r_i,$$

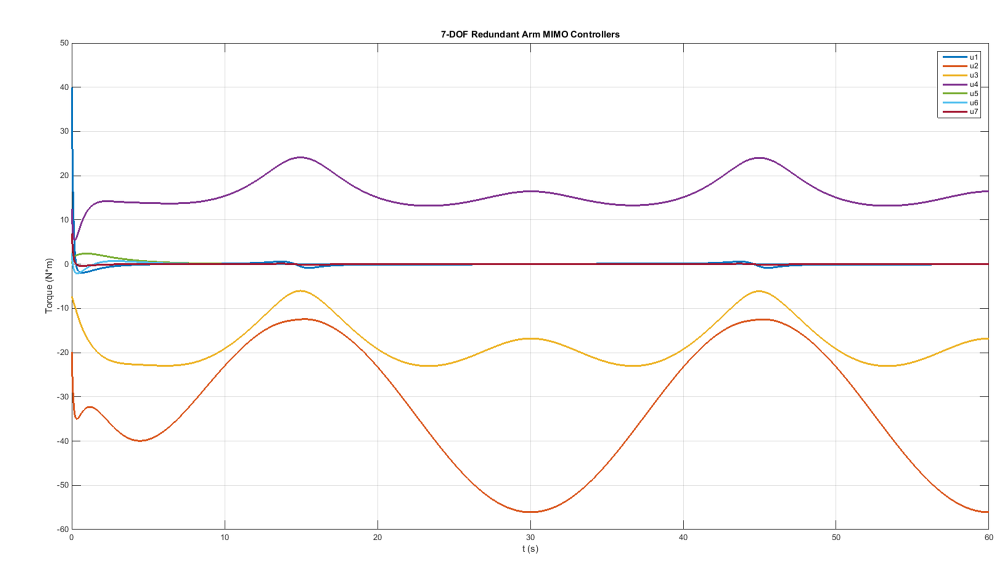
where r_i is end-effector variable reference. We set $v_i = -k_i e_i - k_{i+1}\dot{e}_i$ and get outputs $y_i = -k_i e_i - k_{i+1}\dot{e}_i$.

Performance Applying MIMO Control

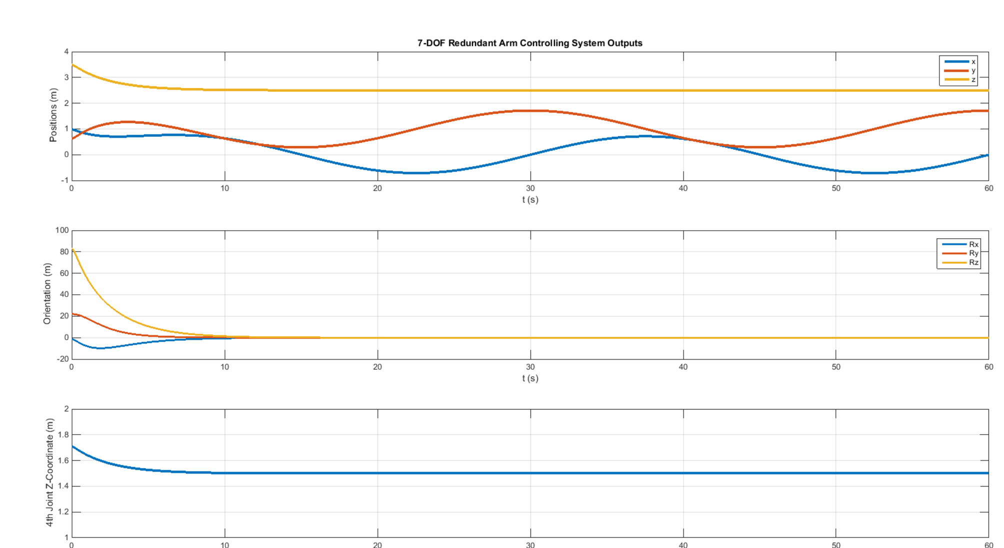
- **Tracking A Circle Reference Path on X-Y Plane**:



- **Input Controllers**:



- **System Outputs**:



Extra Constrains of Redundant System

- **Mathematical Requirement**:
  The $6 \times n$ matrix $G(q_i, \dot{q}_i)$ should be a square matrix in order to derive the Cartesian space MIMO controllers. Therefore, we need to find $n - 6$ more constrain equations for a redundant system (ie. $n > 6$).
- **Additional Constrains**:
  The constrain equations can be chosen differently according to the objective of the redundant robotic arm. One can pick specific geometric limitations of specific joints to avoid obstacles or motion optimizations of the whole arm to save energy, etc.